Remarks

The present response is to the Office Action mailed in the above-referenced case on May 05, 2006. Claims 1-13, 15-22 and 24-32 are standing for examination, and stand rejected under 35 U.S.C. 103(a) as obvious over Jacobson et al. (US 6044402), hereinafter Jacobson, in view of Joiner (US 6742128), hereinafter Joiner, and in view of Carroni et al. (US Application 2002/0143850).

As to claim 1 the Examiner applies Jacobson as the primary reference in the 103 rejection, stating that Jacobson does not teach a hash routing with a sliding window processing, in real time, the data passed over the live connection, but that Joiner teaches the part that Jacobson does not.

The applicant has amended claim 1 in this response to clarify the language somewhat, but the amendment does not alter the essential nature and elements of the claim. Claim 1 as amended recites:

1.(Currently amended) A system for providing network security by managing and manipulating live data connections and connection attempts initiated over a data-packet-network between at least two nodes connected to the network comprising:

a system host machine connected to the network;

a first software application residing on the system host machine for detecting and monitoring the live connections and connection attempts;

a data store for storing data about the live connections and connection attempts;

a second software application for emulating one or more end nodes of the connections or connection attempts; and

a third software application for detecting virus activity by hashing data passed over the live connection in real time and for comparing the hash data to a dataset containing virus signatures, the dataset searchable by hash table index, the hash entries therein derived individually from separate virus signatures;

characterized in that the system using the detection software detects one or more

pre-defined states associated with a particular connection or connection attempt in progress including those associated with any data content or type transferred and performs at least one packet generation and insertion action triggered by the detected state or states, the packet or packets emulating one or more end nodes of the connection or connection attempt to cause preemption or resolution of the detected state or states, and the hashing routine utilizes at least one sliding checksum window processing, in real time, the data passed over the live connection.

The applicant does not agree that the primary reference Jacobson reads on all of the limitations alleged by the Examiner, but the greater fault in the combination is that Joiner does not teach what the Examiner alleges, nor does Carroni teach what the Examiner alleges. Joiner never mentions hashing. Further, Joiner only mentions "sliding window" once, and in a context having nothing to do with claim 1 claim. He says: Ideally, the network data is monitored over a sliding window. In other words, network data is analyzed for a predetermined time period. Quite clearly joiner's "sliding window" refers to a window on a period of time, not a moving view (a "window") of a subset of data in a network packet.

Caronni (2002/1043850) paragraph [0012] and [0029] mentions the concept of hashing virus signatures; however there is no teaching as to how to accelerate the speed of the hash search. This accelerated speed is the goal of the invention.

C hashing (as used for a fast pattern search) works by performing the hash calculation on a short length of data which is a subset of the data being searched, and looking up the hash number in the table. We call this short length of data the "window". As an example, it might be bytes 1 through 9. If the hash value of the first nine bytes is not found in the hash table, then the hashing algorithm is performed on the next "window" – bytes 2 through 10, and the process is repeated. The 9 byte "window" is continually advanced in this manner (3-11, 4-12, 5-13, etc) through the data to be searched for the pattern until the end of the data set (in our case a network packet) is reached. All this is in the prior art. This is the sliding checksum window as claimed.

Our innovation is to avoid the full recalculation of the hash value for each set of window-sized sequential bytes. Instead, we adjust the hash value for the previous window by subtracting (or otherwise "un-hashing") the byte that is passing out of the window (byte 1 in the above example) and adding the value of the next byte entering the window (byte 10 in the above example). This provide about a tenfold increase in hash value calculation efficiency.

None of the patents cited by the examiner address the concept of accelerating hash search speeds by incrementally changing a hash value which is maintained as the "window" advances through the data to be searched. There is no sliding checksum window in the art, nor any motivation to incorporate one.

So the combination fails and claim 1 is patentable over the combination of Jacobson and Joiner. Claims 2-13, 15 and 16, all depending directly or indirectly from claim 1, are therefore patentable at least as depended from a patentable claim.

As to claim 17, the claim is patentable over Jacobson in view of Joiner by the same facts and reasoning stated above on behalf of claim 1. Therefore claims 18-22 and 24-32 are patentable at least as depended from a patentable claim.

As all of applicant's claims are patentable over the prior art, applicant respectfully requests that the rejections be withdrawn and that the case be passed quickly to issue.

If any fees are due beyond fees paid with this amendment, authorization is made to deduct those fees from deposit account 50-0534. If any time extension is needed beyond any extension requested with this amendment, such extension is hereby requested.

Respectfully Submitted,
John Alexander Bartas

By */Donald R/ Boys/*
Donald R. Boys
Reg. No. 35,074

Central Coast Patent Agency
P.O. Box 187
Aromas, CA 95004
(831) 726-1457